REMARKS

The Examiner allowed claims 5, 6, 13, 14, 19, 20, and 27.

The Examiner rejected pending claims 1-4, 7-12, 15-18, and 21-26 as obvious (35 U.S.C. §103) over Pirahesh (U.S. Patent No. 5,548,758) in view of Romer (U.S. Patent No. 5,953,534). . Applicants traverse the prior art rejections for the following reasons.

Independent claims 1, 9, and 17 require transforming data in an input table in a database in a server in communication with a client by: receiving from the client a transform command indicating an input data table name in the database and at least one rule indicating at least one cell in the input table to transform and a transform operation to perform with respect to the at least one cell; accessing a copy of the input table from the database; and transforming, within the server, data in the accessed input table according to each rule specified in the transform command.

The Examiner cited col. 6, line 63 to col. 7, line 6 and col. 5, lines 5-6 of Romer as teaching the claim requirements of receiving from the client a transform command indicating an input data table name in the database and at least one rule indicating at least one cell in the input table to transform and a transform operation to perform with respect to the at least one cell and transforming, within the server, data in the accessed input table according to each rule specified in the transform command. The Examiner recognized that Pirahesh does not teach these requirements and cited Romer to overcome the deficiencies of Romer. (Fourth Office Action, pgs. 2-4) Applicants traverse.

The cited cols. 6-7 of Romer mention that a transformed program may make calls to determine the name and path of executable DLL files. This operation is needed because the name and directory of a module may have changed in the transformed program. According to the cited col. 5, a transformed program has executable modules and DLLs that have different names than the original program. Romer discusses transforming software from an original software program so the transformed software program provides new functionality. (Romer, col. 2, lines 37-56).

Nowhere does the cited Romer anywhere teach or suggest the claim requirement of receiving from the client a transform command indicating an input data table name in the database and at least one rule indicating at least one cell in the input table to transform and a transform operation to perform with respect to the at least one cell. Instead, the cited Romer discusses how a transformed program calls functions to locate a name of a DLL and determine the name and location of modules whose name has changed in the transformed program.

Further, nowhere does the cited Romer anywhere teach or suggest a transform operation to perform on a cell of an input data table indicated in a transform command. The transforms discussed in Romer concern transforms of a program to add functionality to a program. Nowhere does the cited Romer anywhere teach or suggest performing transforms on cells in a data table. In fact, Romer concerns something entirely non-analogous to the claims, transformations of a program and handling any changes to the names of modules that may have changed in the transformed program. Nowhere does the cited Romer anywhere teach or suggest the claim requirements of transformations on a cell in a data table or performing such transformations according to a command indicating an input data table name and at least one rule indicating a cell and a transform to perform. Instead, Romer concerns transforming programs which is different than the field of endeavor of the claims, which concern transformations of cells in a database table.

Thus, the cited Romer does not overcome any of the deficiencies of Pirahesh the Examiner recognized.

The Examiner found that col. 1, lines 12-15 and col. 3, lines 21-41 of Pirahesh teach the claim requirement of transforming data in a database in a server in communication with a client. (Fourth Office Action, pg. 3) Applicants note that Pirahesh generally does not concern transforming data in an input table according to rules as claimed. Instead, Pirahesh concerns how to optimize an SQL query, not transfer data in an input data table in a database. Applicants submit that optimizing an SQL query by transforming a join to an early-out join is a different type of operation than transforming data in an input table as claimed.

The cited col. 1 of Pirahesh discusses database management systems for the optimization of SQL queries using early-out join transformations. The cited col. 3 discusses execution of SQL statements. An application plan for the compiled SQL statements is generated which concern how to get the data the user wants. The cited col. 3 discusses how generating the access plan considers available access paths, such as indexes, sequential reads, etc., and system statistics on the data to access to choose the most efficient access path.

Nowhere do this cited cols. 1 and 3 of Pirahesh anywhere teach or suggest the claim requirement of transforming data in the accessed input table according to each rule specified in the transform command. Instead, the cited Pirahesh concerns how to develop an application plan to execute an SQL query, not how to transform cells in an accessed input table according to rules specified in a transform command. Nowhere does the cited Pirahesh anywhere mention transform commands having rules to transform data in an input table. Instead, the cited Pirahesh concerns something entirely different, how to optimize the execution of an SQL command.

The Examiner further cited the early-out join transformation of Pirahesh as teaching the claimed database transformation. (Fourth Office Action, pgs. 3-4) Applicants submit that the cited join operation is different and does not teach or suggest the claimed transform operation which comprises altering data in cells in an input table. The Application defines the term "transformation" as the process of filtering, merging, decoding, and translating source data to create validated data, such as converting data, applying mathematical or logical operators on the values of data, etc. (Application, col. 2, lines 13-26) The cited join operation of Pirahesh does not transform data in cells in an input table, but instead concatenates rows of data from multiple tables to perform a search with respect to the joined rows.

To further emphasize this distinction, Applicants note that Pirahesh discusses a "transformation" in the context of transforming a join to an early out join to optimize execution of a join query. (Pirahesh, col. 1, lines 54-67). This type of transformation of an SQL query is different than the transformation of data in a cell of a database table as claimed.

Moreover, the claims require that the received transform command indicate at least one cell in the input table to transform and a transform operation to perform with respect to such cell.

The cited join operation of Pirahesh nowhere teaches or suggests the claim requirement of indicating at least one cell in an input table to transform according to an indicated transform operation. Instead, a join operation concerns concatenating rows of a table, not performing transformations on cells in an input table.

Thus, the cited Pirahesh and Romer do not teach or suggest the claim requirements for which they were cited.

Applicants further submit that Pirahesh and Romer are non-analogous references, whose combination does not concern the claimed invention. Pirahesh concerns how to optimize joins in an SQL query, and transforming a join operation to optimize a query. Romer concerns creating a transformed program having new functionality over the original program. Applicants submit that these references are directed to completely different fields of endeavor and concerns. For instance, the cited Romer is concerned with how to determine the name or location of a module that may have been changed in a transformed program and the cited Pirahesh is concerned with transforming a join to an early out join to optimize execution of the join. Accordingly, neither of these references concern the claim requirements of performing transform operations on cells in an input table using a transform operation.

Accordingly, for all the above reasons, Applicants submit that claims 1 9, and 17 are patentable over the cited Pirahesh and Romer, alone or in combination.

Claims 2-4, 7, and 8; 10-16, and 18-21 and 26 are patentable over the cited art because they depend from claims 1, 9, and 17, respectively, which are patentable over the cited art for the reasons discussed above. Moreover, claims 2, 4, 7, 8, 10, 12, 15, 16, 18, 21, 22, and 26 provide additional ground of patentability over the cited art.

Claims 2, 10, and 26 depend from claims 1, 9, and 17 , respectively, and further require that the client is a client computer that communicates with the server over a network, wherein the transform command is transmitted from the client computer to the server over the network. The Examiner cited col. 1, lines 12-13 of Pirahesh as teaching the additional requirements of these claims. (Fourth Office Action, pg. 4) Applicants traverse.

The cited col. 1 of Pirahesh mentions database systems performed by computers and the optimization of queries. Nowhere does this cited col. 1 anywhere teach or suggest the claim requirement that a client computer transmits a transform command as claimed to a server over the network to transform data within the server as claimed.

Accordingly, claims 2, 10, and 26 provide additional grounds of patentability over the cited art.

Claims 4, 12, and 18 depend from claims 1, 9, and 17 and further require that the transform command rules specify multiple transform operations to perform on at least one cell in the accessed input table. An application of a subsequent transform operation following a previous transform operation on one cell transforms previously transformed data in the cell. The Examiner cited the analysis with respect to claim 1 as teaching the additional requirements of these claims. (Fourth Office Action, pg. 5) Applicants traverse.

As discussed above, the cited Pirahesh discusses transformations in the context of transforming a join to an early out join to optimize execution of the join. Again, Applicants submit that joining rows from different tables does not teach or suggest transforming cells in an input table as claimed. The cited Romer concerns how to locate the name and module in a transformed program. Nowhere does the above cited Pirahesh and Romer, alone or in combination, anywhere teach or suggest the claim requirement of rules specifying multiple transform operations on cells in an accessed input table to transform the data, as the term transform is understood.

Accordingly, claims 4, 12, and 19 provide additional grounds of patentability over the cited art.

Claims 7, 15, and 26 depend from claims 1, 9, and 17 and further require that the client cannot affect the execution of the transform command during the execution of the transform command, whereby the transform command executes in the server independently of the client. The Examiner cited the analysis with respect to claim 1 as teaching the additional requirements of these claims. (Fourth Office Action, pg. 5) Applicants traverse.

Applicants submit that the Examiner has not cited any part of Pirahesh or Romer that teaches or suggests the claim requirement that a client cannot affect the execution of the transform command so that the command executes independently of the client that provided the transform command to the server. Applicants request the Examiner to cite specific sections of the references teaching this requirement if the rejection is maintained.

Accordingly, claims 7, 15, and 26 provide additional grounds of patentability over the cited art.

Claims 8, 16, and 21 depend from claims 1, 9, and 17 and further require that the transform command comprises multiple rules, wherein each rule specifies at least one column in the input table and at least one transform operation to perform on each specified column in the input table. At least two rules specify different columns in the input table and different transform operations to apply to each specified column. The Examiner cited the previously cited cols. 6-7 of Romer as teaching the additional requirements of claims 8, 16, and 21. (Fourth Office Action, pgs. 5-6) Applicants traverse.

The cited cols. 6-7 of Romer discusses how a transformed program determines the name or location of DLL files because the name and directory of a called module may have changed in the transformed program. Nowhere does the cited Romer 9 anywhere teach or suggest a transform command having multiple rules, where each rule specifies one column at one transform operation to perform on each specified column, where at least two rules specify different columns in the input table. Instead, the cited col. 9 concerns how a transformed program determines the name of modules it calls.

For these reasons, claims 8, 16, and 21 provide additional grounds of patentability over the cited art.

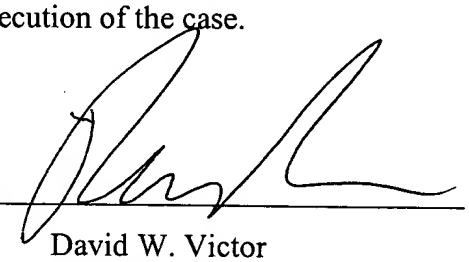
Claims 22-25 include many of the distinguishing requirements found in claims 1, 4, 6, and 8 in data structure format, and are patentable over the cited art for the reasons discussed with respect to claims 1, 4, and 6.

Conclusion

For all the above reasons, Applicant submit that all the pending claims 1-27 are patentable over the art of record. Applicants submit that no fee is needed. Nonetheless, should any additional fees be required, please charge Deposit Account No. 50-0585.

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: August 21, 20031

By: 

David W. Victor
Reg. No.: 39,867

Please direct all correspondences to:

David Victor
Konrad Raynes Victor & Mann, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984